machine learning & knowledge extraction

MDPI

Article

Delving into Unsupervised Hebbian Learning from Artificial Intelligence Perspectives

Wei Lin [1,*], Zhixin Piao [1] and Chi Chung Alan Fung [1,2,*]

[1] Department of Neuroscience, College of Biomedicine, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Kowloon, Hong Kong 999077, China
[2] CityU Shenzhen Research Institute, 8 Yuexing 1st Road, Shenzhen Hi-Tech Industrial Park, Nanshan District, Shenzhen 518057, China
* Correspondence: wlin44-c@my.cityu.edu.hk (W.L.); alan.fung@cityu.edu.hk (C.C.A.F.)

Abstract

Unsupervised Hebbian learning is a biologically inspired algorithm designed to extract representations from input images, which can subsequently support supervised learning. It presents a promising alternative to traditional artificial neural networks (ANNs). Many attempts have focused on enhancing Hebbian learning by incorporating more biologically plausible components. Contrarily, we draw inspiration from recent advances in ANNs to rethink and further improve Hebbian learning in three interconnected aspects. First, we investigate the issue of overfitting in Hebbian learning and emphasize the importance of selecting an optimal number of training epochs, even in unsupervised settings. In addition, we discuss the risks and benefits of anti-Hebbian learning in model performance, and our visualizations reveal that synapses resembling the input images sometimes do not necessarily reflect effective learning. Then, we explore the impact of different activation functions on Hebbian representations, highlighting the benefits of properly utilizing negative values. Furthermore, motivated by the success of large pre-trained language models, we propose a novel approach for leveraging unlabeled data from other datasets. Unlike conventional pre-training in ANNs, experimental results demonstrate that merging trained synapses from different datasets leads to improved performance. Overall, our findings offer fresh perspectives on enhancing the future design of Hebbian learning algorithms.

Keywords: Hebbian learning; overfitting; anti-Hebbian learning; activation functions; hybrid synapses

PACS: 07.05.Mh; 87.19.lj

check for updates

Received: 25 August 2025
Revised: 30 September 2025
Accepted: 6 November 2025
Published: 11 November 2025

Citation: Lin, W.; Piao, Z.; Fung, C.C.A. Delving into Unsupervised Hebbian Learning from Artificial Intelligence Perspectives. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 143. https://doi.org/10.3390/make7040143

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Artificial neural networks (ANNs) powered by backpropagation have demonstrated remarkable performance across various fields [1–3]. However, training ANNs requires significantly more memory than inference does because some optimizers demand additional memory for extra accumulators [4]. Moreover, in visual tasks, the features learned by early convolutional layers closely resemble those found in the early visual processing areas of higher animal brains [5]. To address computational inefficiencies and enhance the biological plausibility of models [6,7], researchers have explored Hebbian learning [8] in the visual field [9–11], aiming to generate meaningful visual representations of images.

A notable contribution in this area is proposed by Krotov et al. [10], who combine Winner-Takes-All (WTA) [12] mechanisms with unsupervised Hebbian learning, achieving

promising results comparable to supervised learning on certain visual datasets. However, this method requires a substantial number of training epochs, making it computationally expensive. To mitigate this issue, many researchers have sought ways to accelerate the training stage for Hebbian learning and have achieved a lot of advances [13,14]. Some researchers even propose that Hebbian learning can achieve acceptable results even with one epoch [11]. In contrast to these prior works, we identify that unsupervised Hebbian learning encounters similar challenges as supervised learning, namely the underfitting and overfitting problems, as shown in Figure 1. To be specific, training Hebbian learning for too many epochs degrades its effectiveness for subsequent training. When training Hebbian learning with only one epoch, the representations obtained by Hebbian learning fail to produce sufficiently informative representations for further training. Our findings based on Fashion-MNIST suggest that training for an optimal number of epochs, normally several epochs, yields the best representations for subsequent supervised learning. Furthermore, we observe that the utilization of anti-Hebbian learning [15,16] can move up the onset of optimal unsupervised learning without decreasing performance in supervised settings. However, anti-Hebbian learning degrades the performance dramatically if it overfits the training data. In this paper, we also visualize trained synapses and find that the synapses trained with the optimal epochs sometimes cannot exhibit perfect images.

Due to anti-Hebbian learning, negative values in trained synapses can result in negative representations in subsequent supervised learning. How to effectively leverage these negative values remains an open question, as many studies on Hebbian learning [9,13,14] rely predominantly on ReLU [17], thereby ignoring the negative components of learned representations. In contrast, many works in ANNs employ a variety of activation functions to tackle this limitation. Motivated by this, we evaluate several commonly used ANN activation functions on representations learned by Hebbian learning. Our extensive experimental results exhibit that Softplus [18] yields the best results for both training and test accuracy among these functions, outperforming GELU [19], which is widely used in popular ANNs [20,21]. Additionally, Swish [22], GELU and LeakyReLU [23] perform better than ReLU. These results indicate that properly leveraging negative values can improve the Hebbian-learned representations for further supervised learning, revealing a partial divergence between Hebbian learning and ANNs in their selection of activation functions.

Building on these previous foundations, we turn to the question of how to leverage more data for Hebbian learning, inspired by the success of pre-training in ANNs [20,24]. In the era of large language models, pre-training on large unlabeled datasets followed by fine-tuning on specific downstream tasks has become a proven and powerful approach [1,24]. Since Hebbian learning is also unsupervised, it naturally thinks about how to leverage irrelevant data to further improve the model's performance. However, in unsupervised Hebbian learning, research mostly focuses on evaluating models trained solely on their respective datasets. To address this, we first propose several data fusion strategies inspired by artificial intelligence algorithms and evaluate their performance. Building on these comparisons, we introduce a novel strategy by independently training weights on two separate unlabeled datasets and then combining them for subsequent supervised learning, and find that it effectively improves performance in the source domain. Our findings reveal a key difference between Hebbian learning and ANNs in strategies for data leveraging.

In this study, our contributions lie in exploring Hebbian learning through three interconnected artificial intelligence perspectives, where insights from each step guide the next. First, we analyze the overfitting issue in unsupervised Hebbian learning and examine the role of anti-Hebbian mechanisms. Next, we conduct extensive experiments to highlight the significance of properly utilizing negative values through activation functions. Finally, we discuss the potential of improving model performance by merging trained weights from

diverse datasets. Collectively, these contributions may offer insightful viewpoints from the perspective of artificial intelligence for advancing the design of Hebbian learning.

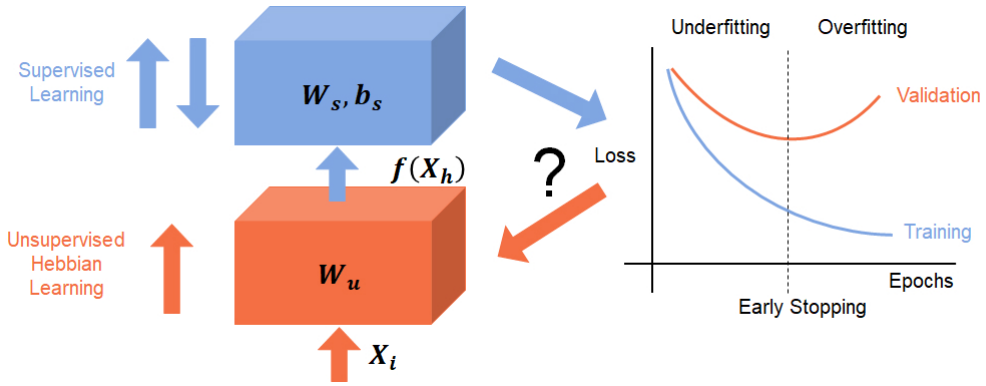

Figure 1. An illustration depicting the workflow of unsupervised Hebbian learning followed by supervised learning, accompanied by a question: Does the overfitting problem commonly found in supervised learning also occur in unsupervised Hebbian learning?

2. Related Work

In the context of neurobiology, Hebbian-like plasticity is associated with long-term potentiation (LTP), while anti-Hebbian plasticity can be interpreted as long-term depression (LTD). Both LTP and LTD are essential for memory processing [25]. In actuality, the global inhibition motif has also been employed in various unsupervised learning algorithms. It is also worth noting that human brains experience similar overfitting issues, necessitating sleep to reset cognitive processes [26]. Besides, overfitting is a well-known challenge in supervised learning [27]. It occurs when, after reaching the optimal training epoch, the training accuracies continue to rise while the test accuracies decline, as shown in Figure 1. In supervised learning, techniques such as dropout [28] and label smoothing [29] are commonly used to prevent overfitting.

Another critical consideration in ANNs is the choice of activation functions. ReLU-based functions that eliminate negative values appear to constrain representational capacity in ANNs [30]. Therefore, activation functions such as LeakyReLU, Softplus, and Gaussian error linear units (GELUs) have been introduced to better utilize negative values and improve performance [20,21,31]. Among these, GELUs are commonly employed in popular pre-trained models [20,24]. Furthermore, variants of gated linear units (GLUs) [32], such as SwiGLU and GEGLU [33], which are based on Swish and GELUs, have become popular in large pre-trained language models [1].

Leveraging irrelevant data for data-limited tasks is an active area of research in ANNs. For instance, BERT [24] pioneered the pre-training paradigm, achieving superior performance on many downstream tasks by learning underlying language structure and providing improved initialization. The huge success of the GPT series [1] similarly follows this approach. This paradigm has extended beyond natural language processing to areas such as computer vision [34,35], molecular graph models [36,37], and protein language models [20,38]. In addition, some studies suggest that incorporating labeled data from varied sources with learning to rank strategies can enhance model performance in the original domain [39]. The model soup technique [40], which averages models trained with different hyperparameters on the same dataset, also achieves better results than models trained under a single configuration.

3. Methodology

In the following section, we employ the classical Hebbian learning algorithm introduced by Krotov et al. [10] to conduct experiments.

3.1. Definitions of Hebbian Learning

3.1.1. Mathematical Formulation of Hebbian Learning

The Winner-Takes-All (WTA) mechanism, combined with Hebbian learning, operates through a two-stage process. For any input image $\mathbf{X_i} \in \mathbb{R}^{N_{in}}$ where $N_{in} = 28 \times 28 = 784$ for MNIST-like datasets, the synaptic weights $\mathbf{W_u} \in \mathbb{R}^{N_{hid} \times N_{in}}$ are used to compute the total input $\mathbf{I}$ to the hidden layer:

$$\mathbf{I} = \text{sign}(\mathbf{W_u}) \odot |\mathbf{W_u}|^{p-1}\mathbf{X_i}, \tag{1}$$

where $\odot$ denotes element-wise multiplication, $p = 2$ is the power parameter, and $\text{sign}(\cdot)$ extracts the sign of each element.

The WTA mechanism determines the activation pattern through ranking the total inputs. For each sample in a batch, the neuron indices are sorted by their total input values: $\mathbf{y} = \text{argsort}(\mathbf{I}, \text{dim} = 0)$. The gating function $g(\mathbf{Q})$ is then defined as

$$g(\mathbf{Q})_i = \begin{cases} 1 & \text{if } i = \text{argmax}(\mathbf{I}) \\ -\Delta & \text{if } i \in \mathbf{k} \text{ (anti-Hebbian indices)} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where Δ controls the strength of anti-Hebbian learning and $\mathbf{k}$ specifies which ranked neurons receive anti-Hebbian updates.

3.1.2. Hebbian Weight Update Rule

The synaptic weight updates follow a normalized Hebbian learning rule. For each training step, the weight change $\Delta\mathbf{W}$ is computed as

$$\Delta\mathbf{W} = \frac{\epsilon}{N_c}\Big(g(\mathbf{Q})\mathbf{X_i}^T - \text{diag}(g(\mathbf{Q}) \cdot \mathbf{I})\mathbf{W_u}\Big), \tag{3}$$

where ϵ is the learning rate that decreases linearly from $\epsilon_0 = 0.02$ to 0 over the training epochs, $N_c = \max(|\Delta\mathbf{W}|, 10^{-30})$ is the normalization factor to prevent numerical instability, and $\text{diag}(\cdot)$ creates a diagonal matrix. The second term on the right-hand side serves as a relaxation of the dynamics. This relaxation can be justified by an implicit homeostatic mechanism that regularizes the synaptic connection.

The learning rate schedule is implemented as

$$\epsilon(t) = \epsilon_0\left(1 - \frac{t}{T}\right), \tag{4}$$

where t is the current epoch and T is the total number of training epochs.

3.1.3. Implementation of Anti-Hebbian Mechanism

The anti-Hebbian mechanism is implemented by applying negative updates to specific ranked neurons. In our implementation, we use $k = 7$, meaning the 7th-ranked neuron (i.e., $N_{hid} - 7$) receives the anti-Hebbian update with strength $-\Delta$. This creates competition between the winning neuron (rank 1) and the anti-Hebbian neuron, promoting sparse and diverse representations.

3.2. Supervised Learning Settings

After deploying unsupervised Hebbian learning, the learned representations are used for supervised classification. The hidden layer activations $\mathbf{X_h}$ are computed as

$$\mathbf{X_h} = f(\mathbf{W_u X_i}), \tag{5}$$

where $f(\cdot)$ is the activation function. These representations are then fed into a single fully connected layer for classification:

$$\mathbf{X_o} = \text{softmax}(\mathbf{W_s X_h} + \mathbf{b_s}), \tag{6}$$

where $\mathbf{W_s} \in \mathbb{R}^{10 \times N_{hid}}$ and $\mathbf{b_s} \in \mathbb{R}^{10}$ are the classification weights and biases, respectively.

3.3. Definition of Different Activation Functions

In this study, we aim to introduce some alternative activation functions, such as LeakyReLU, GELU, Swish, and Softplus, to investigate how to better exploit negative values in representations learned by Hebbian learning. The differing behaviors of four activation functions, namely ReLU, LeakyReLU, GELU, Swish, and Softplus, near zero are illustrated in Figure 2. In addition, the rectified polynomial unit (RePU) [41], originally proposed in brain-inspired models as part of an energy function, is also examined in the following section.

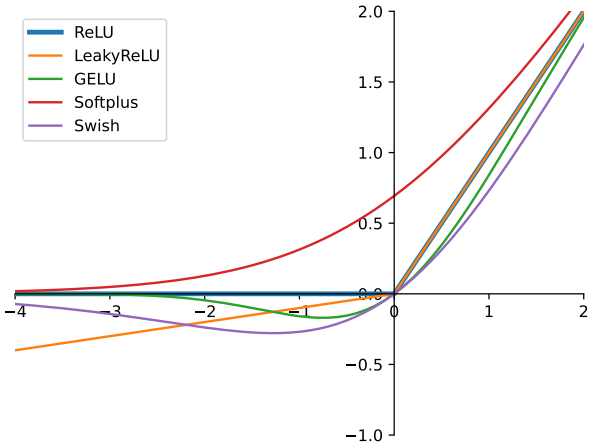

Figure 2. Visualization of ReLU, LeakyReLU, GELU, Swish, and Softplus around zero.

The mathematical formulation of ReLU is given by Equation (7).

$$\text{ReLU}(x) = \max(0, x) \tag{7}$$

LeakyReLU, formulated in Equation (8), introduces a negative slope n_s for input values less than zero. Here, we set $n_s = 0.1$ to evaluate the contribution of negative values.

$$\text{LeakyReLU}(x) = \max(n_s x, x) \tag{8}$$

GELU is connected to stochastic regularizers in that it is the expected value of a modified form of adaptive dropout [42]. The formulation of GELU is shown in Equation (9).

$$\text{GELU}(x) = \frac{x}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{9}$$

The mathematical formulation of Swish is shown in Equation (10).

$$\text{Swish}(x) = x \cdot \sigma(x) \tag{10}$$

The formulation of Softplus is illustrated in Equation (11).

$$\text{Softplus}(x) = \log(1 + e^x) \tag{11}$$

The formulation of RePU is defined in Equation (12), where $n > 1$. In this study, we set $n = 4.5$, following previous work [10].

$$\text{RePU}(x) = \max(0, x)^n \tag{12}$$

3.4. Statistics of Datasets

Since MNIST is an easy dataset for training which allows the training accuracy to approach nearly 100%, we conducted experiments with a more challenging dataset, namely Fashion-MNIST [43], to avoid this issue. Additionally, MNIST and KMNIST (Kuzushiji-MNIST) [44] were used as the assistant datasets to verify the feasibility of utilizing other datasets. The details of these datasets are displayed in Table 1.

Table 1. Statistics and descriptions of different datasets.

Dataset	Training Samples	Testing Samples	Description
Fashion-MNIST	60,000	10,000	Ten types of clothing
MNIST	60,000	10,000	Ten types of handwritten digits
KMNIST	60,000	10,000	Ten types of handwritten digits in Japanese

3.5. Approaches to Data Fusion

3.5.1. Strategies for Utilizing Additional Unlabeled Datasets

Drawing inspiration from pre-training methods [24,36] and hybrid data training approaches [39], we propose several data fusion strategies (denoted as s1–s7) to evaluate against the performance of synapses trained solely on the original dataset (see s0).

Diverse strategies for utilizing datasets:

- s0. Synapses are trained on the Fashion-MNIST training set for 25 epochs.
- s1. Synapses are first trained on the KMNIST training set for 5 epochs, followed by training on the Fashion-MNIST training set for 20 epochs.
- s2. Two separate sets of synapses are independently trained for 25 epochs, one on the Fashion-MNIST training set and the other on the KMNIST training set, before being combined.
- s3. Two separate sets of synapses are independently trained for 25 epochs, one on the Fashion-MNIST training set and the other on the MNIST training set, before being combined.
- s4. Two separate synapses are independently trained for 25 epochs, one on the Fashion-MNIST training set and the other on a combined set of KMNIST and MNIST training data, before being combined.
- s5. Two separate synapses are independently trained for 25 epochs, one on the Fashion-MNIST training set and the other on its test set, before being combined.
- s6. Synapses are trained for 25 epochs on the combined Fashion-MNIST training and test sets.
- s7. Synapses are trained for 20 epochs on the combined Fashion-MNIST training and test sets.

3.5.2. Equation for Synaptic Weight Merging with Multiple Datasets

For leveraging multiple datasets, we propose a synaptic weight merging strategy. Two sets of synaptic weights are independently trained on different datasets for the same number of epochs, then combined using the following equation:

$$\mathbf{W_u} = (1 - \alpha)\mathbf{W_u}^s + \alpha\mathbf{W_u}^o \tag{13}$$

where $\mathbf{W_u}^s$ represents weights trained on the source domain (Fashion-MNIST), $\mathbf{W_u}^o$ represents weights trained on the auxiliary domain (MNIST, KMNIST, etc.), and α controls the hybrid ratio. A diagram showing the merging of two synapses trained on different datasets for further supervised learning is shown in Figure 3, where the combined synapses $\mathbf{W_u}$ are used for the subsequent supervised learning.

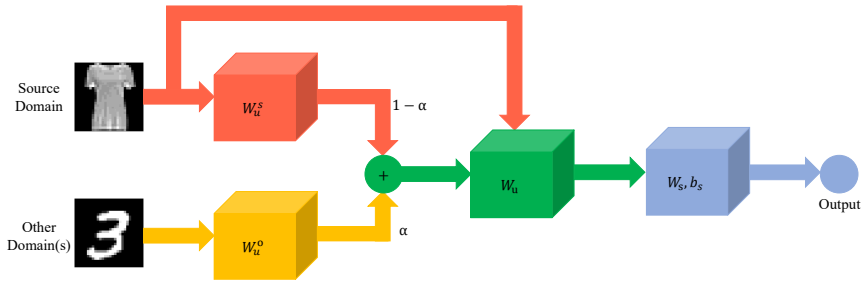

Figure 3. Diagram illustrating the merging of synapses W_u^s trained on the source domain and synapses W_u^o trained on the other domain, prior to supervised learning on the source domain.

3.6. Training Details

We set the number of neurons to match the number of pixels in each image for consistency with the visualizations. Specifically, this meant using 784 neurons for Fashion-MNIST. Specifically, we selected the values of $p = 2$ and $k = 7$, which are near the two sets of optimal hyperparameters, as well as a set that fails to converge on the MNIST dataset reported in the previous study [10].

Unsupervised-phase hyperparameters:

- Batch size: 100.
- Learning rate: Linear decay from $\epsilon_0 = 0.02$ to 0.
- Number of hidden neurons: $N_{hid} = 784$ (matching input dimensionality).
- Power parameter: $p = 2$.
- Anti-Hebbian strength: $\Delta = 0.4$.
- Anti-Hebbian indices: $k = 7$.
- Numerical stability threshold: prec $= 10^{-30}$.

Supervised-phase hyperparameters:

- Batch size: 100.
- Learning rate: Cosine annealing from 0.01 to 0 over 200 epochs.
- Optimizer: Adam [4].
- Loss function: Cross-entropy.

Weight initialization: Synaptic weights are initialized as a normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$.

Experimental setup: All experiments were conducted with multiple random seeds to ensure statistical reliability. The Fashion-MNIST dataset served as the primary evaluation domain, while MNIST and KMNIST were used as auxiliary datasets for the multi-domain experiments. Each experiment was repeated three times with different random seeds, and results are reported as means ± standard deviation.

4. Results

In this study, our goals are to identify the optimal number of training epochs for unsupervised Hebbian learning, the role of anti-Hebbian learning, the impact of various activation functions and strategies for leveraging other datasets to enhance model performance in the source domain.

4.1. Overfitting in Hebbian Learning

Unlike supervised learning, which requires both forward and backward propagation, unsupervised Hebbian learning relies solely on forward propagation. Figure 1 illustrates the overall framework of unsupervised Hebbian learning, which consists of two stages: unsupervised Hebbian learning for extracting image representations and supervised learning for classification tasks.

4.1.1. Degraded Performance Caused by Overfitting

As exhibited in Figure 4, the highest training and test accuracy is achieved when the unsupervised epoch is 20 for $\Delta = 0.4$. In contrast, with $\Delta = 0.0$, the best training accuracy occurs at epoch 10, while the peak test accuracy is reached at epoch 50. Compared with these results for $\Delta = 0.0$, $\Delta = 0.4$ leads to better performance for training accuracy without compromising the test accuracy, and does so with fewer unsupervised epochs. However, anti-Hebbian learning acts as a double-edged sword. Although it can boost performance with fewer epochs, it also causes a significant drop in both training and test accuracy when overfitting occurs, especially with unsupervised epochs set to 50 or 100. This decline is less pronounced in the absence of anti-Hebbian learning.

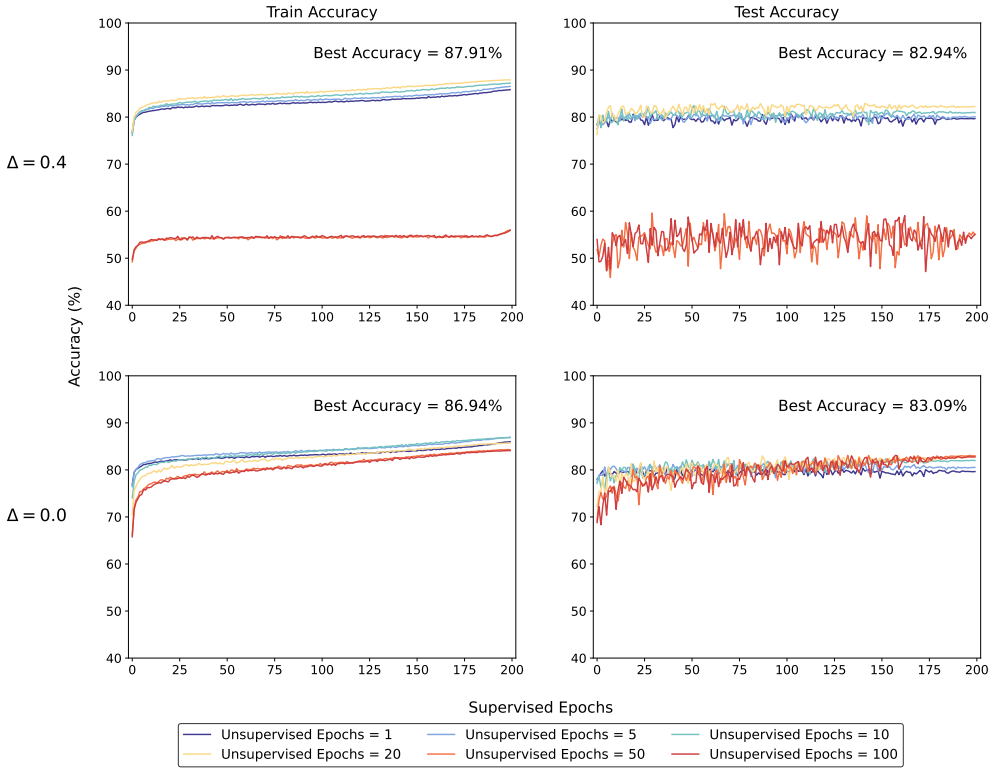

Figure 4. Model performance of unsupervised Hebbian learning on the Fashion-MNIST dataset, evaluated across different values of Δ and number of training epochs, showing average accuracy over three runs.

Another noteworthy phenomenon is underfitting, observed when comparing the optimal epoch to training epochs shorter than the optimal one. For instance, when $\Delta = 0.0$

or $\Delta = 0.4$, both training accuracy and test accuracy at epoch 1 perform worse than those at the optimal epoch, suggesting the necessity of training for several epochs in unsupervised Hebbian learning.

4.1.2. Visualization of Trained Synapses Under Overfitting

To further investigate the underlying mechanism of unsupervised Hebbian learning, we visualize the trained synapses at different epochs, as shown in Figure 5. For $\Delta = 0.4$, the trained synapses begin to exhibit the rough outline of the images at epoch 20. However, when overfitting occurs, anti-Hebbian learning degrades the visualization quality of the trained synapses when the training epoch is set to 50, whereas the visualization quality improves without the use of anti-Hebbian learning. For $\Delta = 0.0$ at epoch 50, which corresponds to its optimal hyperparameter setting for the test accuracy, the synapses fit the input images. Nevertheless, this alignment does not occur at the epoch that yields the best training accuracy. Together, these findings reveal that synapses trained to achieve the best model performance do not always visually represent the input images accurately.

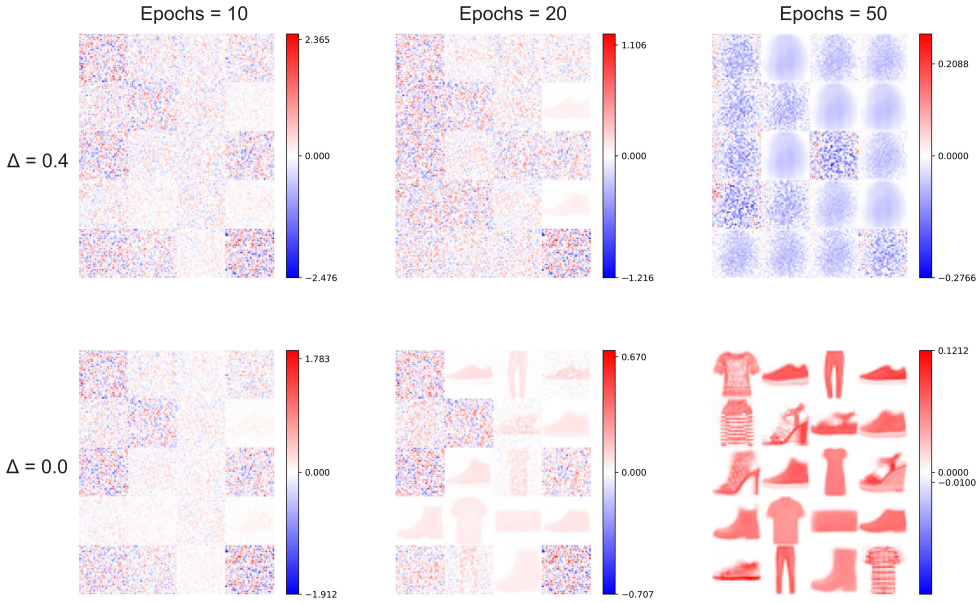

Figure 5. Visualizations of synapses trained by unsupervised Hebbian learning on Fashion-MNIST, evaluated under different Δ values and epochs.

4.2. Effects of Activation Functions on Representations Learned by Hebbian Learning

Based on prior experience, we narrow the range of training epochs and select training epochs of 20, 25, and 30 for evaluation. In addition, we used anti-Hebbian learning with the hyperparameter of $\Delta = 0.4$.

According to the results in Table 2, RePU, ReLU, and GELU all achieve their highest performance at epoch 20, but their accuracy significantly drops by epoch 30, indicating that activation functions do not affect the overfitting problem in Hebbian learning. Notably, the polynomial order in RePU does not yield further improvements in training and test accuracy compared with ReLU. In contrast to these three activation functions, both LeakyReLU and Softplus demonstrate substantial improvements over ReLU and achieve their best results at epoch 25, showing the influence of activation function on the optimal training epoch. In contrast, GELU provides only marginal improvements of 0.05% and 0.01%, respectively. The findings indicate the usefulness of retaining negative values. Among these activation functions, Softplus delivers the best overall performance across all settings, notably improving training accuracy by 1.63% and test accuracy by 0.5% compared to ReLU.

However, Softplus is sensitive to the overfitting issue, as evidenced by a sharp performance decline between epochs 25 and 30, which is less pronounced with ReLU, LeakyReLU, Swish, and GELU.

Table 2. Average accuracy (over three runs) of supervised learning trained with diverse activation functions across different epochs. The best accuracy for each activation function across epochs is underlined, while the overall best accuracy among all conditions is highlighted in bold.

Metric	**Training Accuracy**			**Test Accuracy**		
Epochs	**20**	**25**	**30**	**20**	**25**	**30**
RePU	87.91 ± 0.10	77.84 ± 2.29	56.08 ± 0.05	82.25 ± 0.05	73.02 ± 2.07	54.91 ± 0.32
ReLU	89.27 ± 0.05	85.78 ± 2.02	67.08 ± 0.04	84.57 ± 0.04	80.01 ± 2.18	66.35 ± 0.11
LeakyReLU	89.15 ± 0.04	90.21 ± 0.34	86.35 ± 0.05	84.53 ± 0.05	84.66 ± 0.20	84.03 ± 0.07
GELU	89.32 ± 0.04	87.25 ± 0.85	68.62 ± 0.07	84.58 ± 0.08	81.81 ± 0.93	67.02 ± 0.13
Swish	89.44 ± 0.09	89.18 ± 0.61	75.31 ± 1.46	84.67 ± 0.17	83.57 ± 0.69	72.95 ± 1.21
Softplus	89.46 ± 0.10	**90.90 ± 0.26**	72.24 ± 0.12	84.68 ± 0.27	**85.07 ± 0.46**	70.42 ± 0.11

4.3. Results of Data Fusion Strategies

In this work, inspired by the paradigms of pre-training [24,36] or hybrid data training [39], we first designed seven strategies for utilizing other datasets (see s1–s7 in Section 3.5.1) and then compared their performance on the Fashion-MNIST test set. Notably, we utilize anti-Hebbian learning and Softplus as the activation function, inspired by previous experience.

4.3.1. Comparison of Different Data Fusion Strategies

The results across eight different scenarios are presented in Figure 6. We observe that combining two separately trained sets of synapses, one trained on the Fashion-MNIST training set and the other on an external dataset, consistently outperforms training solely on the Fashion-MNIST training set (see s2–s5). Among these, using MNIST as the secondary dataset yields the best performance, surpassing combinations involving KMNIST, the combined KMNIST and MNIST training sets, or the Fashion-MNIST test set. Intriguingly, incorporating synapses trained on the Fashion-MNIST test set for Hebbian learning (see s5) does not bring significant improvements as might be expected, indicating the differences between supervised learning and Hebbian learning. Notably, merging the training and test sets for Hebbian learning with an optimal training epoch (see s7) yields some improvements compared with training solely on the training set of Fashion-MNIST, but this strategy performs worse than the previous strategy with the test set of Fashion-MNIST (see s5). As the number of training samples increases, the optimal training epoch shifts upward (compare s6 with s7). Conversely, pre-training on external datasets, namely KMNIST, before fine-tuning on the training set of Fashion-MNIST with Hebbian learning leads to a significant drop in performance (see s1).

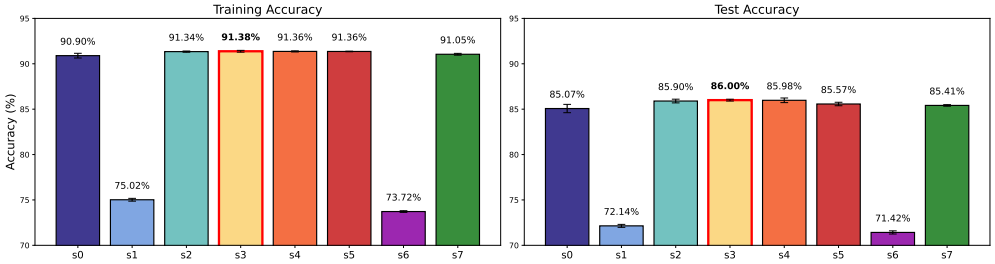

Figure 6. Training accuracy and test accuracy on Fashion-MNIST with eight data leveraging strategies, labeled s0 through s7.

4.3.2. Influence of Hybrid Parameter

To further investigate the influence of the hybrid parameter α, we conducted experiments varying α within the range $[0.0, 0.5]$. The results, shown in Figure 7, indicate that hybrid synapses outperform those trained only on Fashion-MNIST when α lies between 0.1 and 0.3. Furthermore, for α values in the broader range $[0.1, 0.5]$, the hybrid synapses consistently achieve better performance than synapses trained solely on Fashion-MNIST. The optimal performance of hybrid synapses occurs at $\alpha = 0.1$, yielding improvements of 0.5% in training accuracy and 0.92% accuracy compared to training on the original dataset. These results exhibit the usefulness and generalizability of our strategy for hybrid synapses trained on different datasets.

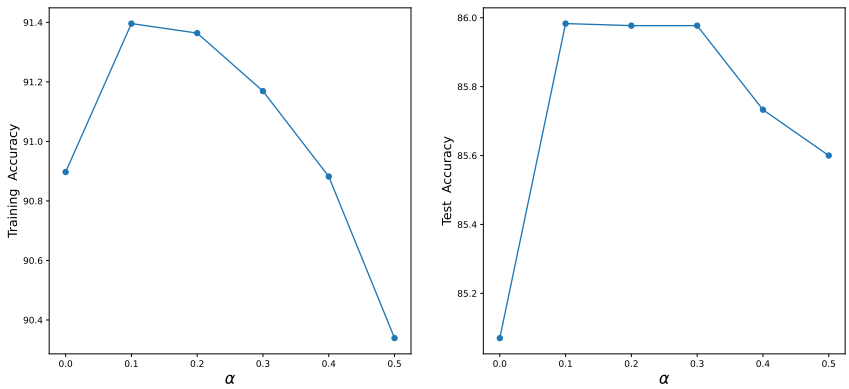

Figure 7. Training and test accuracy on the test set of Fashion-MNIST under varying α, with synapses independently trained on the training sets of Fashion-MNIST and MNIST.

5. Conclusions

In this study, we begin by examining overfitting in the context of unsupervised Hebbian learning, conducting extensive experiments on the Fashion-MNIST dataset. Our findings reveal that, similar to supervised learning, there exists an optimal number of training epochs, beyond which performance declines. We further explore the potential benefits and limitations of incorporating anti-Hebbian mechanisms into unsupervised Hebbian learning, finding that while anti-Hebbian learning can advance the onset of optimal unsupervised learning without harming performance in supervised settings, it causes significant degradation when overfitting occurs. Additionally, our visualizations indicate that synapses resembling input images do not necessarily reflect effective learning.

Building on insights from artificial intelligence, we assess the impact of different activation functions, focusing on how negative values within representations learned by Hebbian learning influence learning performance. Our experiments demonstrate that Softplus outperforms alternatives such as ReLU, RePU, LeakyReLU, Swish, and GELU, primarily because its ability to preserve and utilize negative values enables the generation of more nuanced representations. Besides, using Softplus and LeakyReLU tends to delay the optimal training epoch. These findings underscore the critical importance of leveraging negative representations learned by Hebbian learning.

Finally, inspired by the success of pre-training, we investigate the feasibility of incorporating multi-domain unlabeled data to enhance performance in unsupervised Hebbian learning systems. Drawing inspiration from pre-training or hybrid data training, we develop and evaluate several novel approaches for leveraging data from multiple domains in the Hebbian learning context. Our systematic exploration reveals significant and counterintuitive effects when multi-domain data is incorporated into unsupervised Hebbian networks. The most striking finding is that optimal performance is achieved through a synaptic weight merging strategy, where separately trained synapses from different

datasets are combined post-training. This approach demonstrates substantial improvements over single-domain baselines. Conversely, conventional multi-domain strategies such as pre-training on external datasets or jointly training on combined datasets from multiple domains significantly degrade performance, highlighting unique characteristics of how Hebbian learning systems process multi-domain information.

6. Discussion

In supervised learning, various techniques such as dropout and label smoothing are commonly employed to prevent overfitting. In this study, we exhibit that anti-Hebbian learning and the choice of appropriate activation functions serve similar roles in mitigating overfitting within Hebbian learning. This suggests that future Hebbian learning designs could benefit from incorporating more ideas inspired by artificial neural networks.

Our results highlight Softplus as a particularly effective activation function for processing representations learned via Hebbian learning, indicating that its potential may be underestimated in conventional ANN design. Moreover, Softplus may exhibit even greater promise when combined with gated linear units for large pre-trained models compared with GELU.

Although our method achieves improved performance, it is worth incorporating additional concepts from artificial intelligence, such as self-supervised learning [45], to attempt the feasibility of data fusion strategies. The findings on utilizing additional datasets reveal fundamental differences between Hebbian learning and conventional ANNs in their capacity to leverage multi-domain data effectively and provide many insights for different directions. First, adding noise to the embeddings of large language models results in increased training loss but decreased test loss [46]. This parallels the observation that the hybrid parameter has a narrow optimal range for training accuracy improvement, yet a wide range for boosting test accuracy. Furthermore, this may explain why, when $\Delta = 0.0$, the synapses achieving the best test accuracy at epoch 50 fit the input images well, while those at the epoch with the best training accuracy do not. Notably, this data fusion strategy can achieve better performance with only a single set of trained weights, highlighting its potential for resource-constrained hardware [47]. Currently, our work focuses on single-channel datasets. Extending the multi-domain strategy to multi-channel datasets using convolutional neural networks [48] represents a promising direction for our future research. Furthermore, given the diversity of unsupervised brain-inspired algorithms [49], including forward–forward learning [50], our multi-domain merging strategy may prove applicable to these alternative unsupervised learning frameworks as well.

Author Contributions: Conceptualization, W.L. and C.C.A.F.; methodology, W.L.; software, W.L.; validation, W.L. and Z.P.; formal analysis, W.L.; investigation, W.L.; data curation, Z.P.; writing—original draft preparation, W.L.; writing—review and editing, W.L., Z.P. and C.C.A.F.; visualization, W.L. and Z.P.; supervision, C.C.A.F.; project administration, C.C.A.F.; funding acquisition, C.C.A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by a start-up grant (grant no.: 9610591) for New Faculty and internal grants (grant nos.: 7006055, 7020162, 9680367) from the City University of Hong Kong to C.C.A.F., and a general grant (project no.: JCYJ20230807115001004) from the Science, Technology and Innovation Commission of Shenzhen Municipality to C.C.A.F. (the Shenzhen Research Institute, City University of Hong Kong).

Data Availability Statement: The data used in this study can be accessed at https://github.com/pytorch/vision/blob/main/torchvision/datasets/, accessed on 1 July 2025.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774. [CrossRef]
2. Sejnowski, T.J. The unreasonable effectiveness of deep learning in artificial intelligence. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 30033–30038. [CrossRef]
3. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
4. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
5. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part I 13 ; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.
6. Bengio, Y.; Lee, D.H.; Bornschein, J.; Mesnard, T.; Lin, Z. Towards biologically plausible deep learning. *arXiv* **2015**, arXiv:1502.04156.
7. Illing, B.; Gerstner, W.; Brea, J. Biologically plausible deep learning—But how far can we go with shallow networks? *Neural Netw.* **2019**, *118*, 90–101. [CrossRef]
8. Hebb, D.O. *The Organization of Behavior: A Neuropsychological Theory*; Psychology Press : London, UK, 2005.
9. Fung, C.C.A.; Fukai, T. Competition on presynaptic resources enhances the discrimination of interfering memories. *PNAS Nexus* **2023**, *2*, pgad161. [CrossRef]
10. Krotov, D.; Hopfield, J.J. Unsupervised learning by competing hidden units. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 7723–7731. [CrossRef]
11. Moraitis, T.; Toichkin, D.; Journé, A.; Chua, Y.; Guo, Q. Softhebb: Bayesian inference in unsupervised hebbian soft winner-take-all networks. *Neuromorphic Comput. Eng.* **2022**, *2*, 044017. [CrossRef]
12. Maass, W. On the computational power of winner-take-all. *Neural Comput.* **2000**, *12*, 2519–2535. [CrossRef] [PubMed]
13. Gupta, M.; Ambikapathi, A.; Ramasamy, S. Hebbnet: A simplified hebbian learning framework to do biologically plausible learning. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021 ; pp. 3115–3119.
14. Lagani, G.; Falchi, F.; Gennaro, C.; Fassold, H.; Amato, G. Scalable bio-inspired training of Deep Neural Networks with FastHebb. *Neurocomputing* **2024**, *595*, 127867. [CrossRef]
15. Földiak, P. Forming sparse representations by local anti-Hebbian learning. *Biol. Cybern.* **1990**, *64*, 165–170.
16. Pehlevan, C.; Hu, T.; Chklovskii, D.B. A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data. *Neural Comput.* **2015**, *27*, 1461–1495. [CrossRef]
17. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
18. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011 ; pp. 315–323.
19. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
20. Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C.L.; Ma, J.; et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2016239118. [PubMed]
21. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
22. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941. [CrossRef]
23. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the International Conference on Machine Learning , Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.
24. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
25. Malenka, R.C.; Bear, M.F. LTP and LTD. *Neuron* **2004**, *44*, 5–21. [CrossRef] [PubMed]
26. Hoel, E. The overfitted brain: Dreams evolved to assist generalization. *Patterns* **2021**, *2*, 100244 . [CrossRef] [PubMed]
27. Salman, S.; Liu, X. Overfitting mechanism and avoidance in deep neural networks. *arXiv* **2019**, arXiv:1901.06566. [CrossRef]
28. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
29. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 18–20 June 2016; pp. 2818–2826.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 18–20 June 2016; pp. 770–778.

31. Lin, W.; Fung, C.C.A. Utilizing Data Imbalance to Enhance Compound–Protein Interaction Prediction Models. *Adv. Intell. Syst.* **2025**, 2400985 . [CrossRef]
32. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 933–941.
33. Shazeer, N. Glu variants improve transformer. *arXiv* **2020**, arXiv:2002.05202. [CrossRef]
34. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009.
35. Bao, H.; Dong, L.; Piao, S.; Wei, F. BEiT: BERT Pre-Training of Image Transformers. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021.
36. Lin, W.; Fung, C.C.A. Rethinking the Masking Strategy for Pretraining Molecular Graphs from a Data-Centric View. *ACS Omega* **2024**, *9*, 20832–20838. [CrossRef]
37. Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.S.; Leskovec, J. Strategies for Pre-training Graph Neural Networks. In Proceedings of the 8th International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia, 26–30 April 2020 .
38. Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rehawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M.; et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7112–7127.
39. Luo, H.; Xiang, Y.; Fang, X.; Lin, W.; Wang, F.; Wu, H.; Wang, H. BatchDTA: Implicit batch alignment enhances deep learning-based drug–target affinity estimation. *Briefings Bioinform.* **2022**, *23*, bbac260. [CrossRef]
40. Wortsman, M.; Ilharco, G.; Gadre, S.Y.; Roelofs, R.; Gontijo-Lopes, R.; Morcos, A.S.; Namkoong, H.; Farhadi, A.; Carmon, Y.; Kornblith, S.; et al. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 23965–23998.
41. Krotov, D.; Hopfield, J.J. Dense associative memory for pattern recognition. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16), Red Hook, NY, USA, 5–10 December 2016; pp. 1180–1188.
42. Ba, J.; Frey, B. Adaptive dropout for training deep neural networks. In Proceedings of the Advances in Neural Information Processing Systems 26 (NIPS 2013), Lake Tahoe, NV, USA, 5–8 December 2013; Volume 26 .
43. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747. [CrossRef]
44. Clanuwat, T.; Bober-Irizar, M.; Kitamoto, A.; Lamb, A.; Yamamoto, K.; Ha, D. Deep learning for classical japanese literature. *arXiv* **2018**, arXiv:1812.01718. [CrossRef]
45. Wu, J.; Pan, Y.; Ye, Q.; Zhou, J.; Gou, F. Intelligent cell images segmentation system: Based on SDN and moving transformer. *Sci. Rep.* **2024**, *14*, 24834. [CrossRef] [PubMed]
46. Jain, N.; Chiang, P.y.; Wen, Y.; Kirchenbauer, J.; Chu, H.M.; Somepalli, G.; Bartoldson, B.R.; Kailkhura, B.; Schwarzschild, A.; Saha, A.; et al. NEFTune: Noisy Embeddings Improve Instruction Finetuning. In Proceedings of the ICLR, Vienna, Austria, 7–11 May 2024.
47. Lv, Z.; Zhu, S.; Wang, Y.; Ren, Y.; Luo, M.; Wang, H.; Zhang, G.; Zhai, Y.; Zhao, S.; Zhou, Y.; et al. Development of bio-voltage operated humidity-sensory neurons comprising self-assembled peptide memristors. *Adv. Mater.* **2024**, *36*, 2405145. [CrossRef] [PubMed]
48. Journé, A.; Rodriguez, H.G.; Guo, Q.; Moraitis, T. Hebbian deep learning without feedback. *arXiv* **2022**, arXiv:2209.11883.
49. Schmidgall, S.; Ziaei, R.; Achterberg, J.; Kirsch, L.; Hajiseyedrazi, S.; Eshraghian, J. Brain-inspired learning in artificial neural networks: A review. *APL Mach. Learn.* **2024**, *2*, 021501. [CrossRef]
50. Hinton, G. The forward-forward algorithm: Some preliminary investigations. *arXiv* **2022**, arXiv:2212.13345. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.